Використання налагоджувана програм у візуальному середовищі програмування. Покрокове виконання програм, перегляд значень змінних під час виконання програми. Різновиди помилок, методи їх пошуку та виправлення

Написати працездатну складну програму з першого разу дуже непросто. Звичайно, середовище Delphi допоможе у виправленні синтаксичних помилок, однак припуститися неточності або помилки можна й у формально правильно записаному операторі. У подібних випадках нам допоможе режим налагодження та покрокового виконання, в якому програму можна виконати оператор за оператором, аналізуючи значення проміжних змінних. У цьому розділі ми розглянемо технічні можливості налагодження додатків у середовищі Delphi, ознайомимося з методикою виправлення помилок.

Кваліфікацію програміста визначає не його вміння писати безпомилкові програми (написати складну програму без помилок практично неможливо). Майстерність визначається вмінням розробника швидко, ефективно й надійно налагоджувати і тестувати свій додаток.

Компіляція з дальшим виконанням додатка здійснюється за допомогою команди **Run** (в основному меню) → **Run** або «гарячої» клавіші **F9** (або вибір ▶). Виконання буде здійснюватися тільки у випадку, якщо в ході компіляції не виявлено помилок і завантажувальний модуль створено. Компіляція без дальшого виконання додатка здійснюється за допомогою команди **Project Compile Project** або «гарячих» клавіш **Ctrl + F9**. У результаті компіляції можуть бути зроблені зауваження — **Hint**, попередження — **Warning** і повідомлення про помилки — **Error**. Наприклад, у лістингу програми, поданої на рис. 6.1, є рядок із попередженням про необхідність використати як параметр циклу **FOR** локальну змінну та рядок із повідомленням про помилку невідповідності типів **Integer** і **Extended**.

Якщо наш додаток відкомпілювався і став виконуватися, це, на жаль, іще не означає, що він правильно працює. У ньому може бути ще безліч помилок, які даватимуться взнаки під час виконання додатка. Це можуть бути постійні помилки, які весь час виникають у ході виконання певних частин нашої програми, або помилки, що виявляються тільки у випадку якихось сполучень даних: помилки ділення на нуль, переповнення, відкриття неіснуючого файла тощо. Нарешті, можуть бути випадкові перемежовані помилки, коли одна й та сама задача іноді виконується нормально, а іноді — ні. Такі помилки, які найважчевиявити, зазвичай пов'язані з відсутністю ініціалізації змінних у різних режимах роботи. У

цьому випадку виконання додатка залежить від випадкового стану пам'яті комп'ютера.

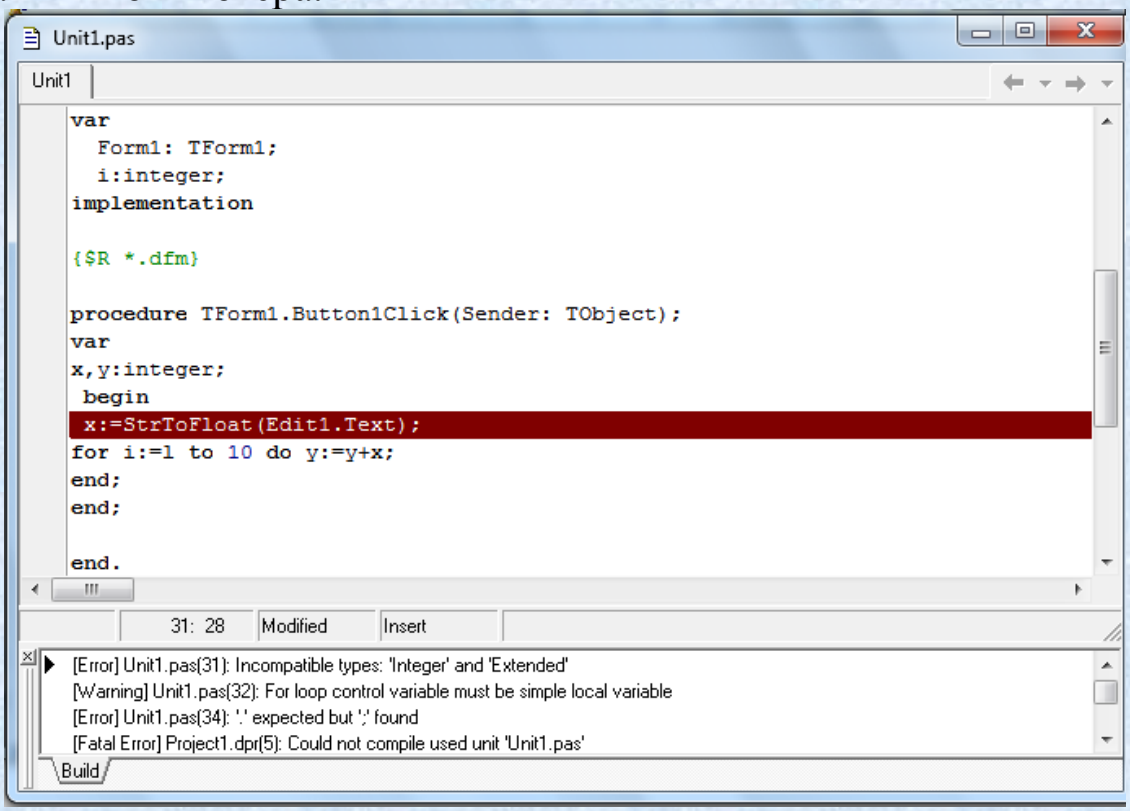

Рис. 6.1. Лістинг програми, що містить попередження про помилку

Дуже часто для того, щоб з'ясувати, у чому причина помилки, необхідно, виконуючи фрагмент програми, відстежувати, як змінюються змінні під час виконання кожної команди. Для покрокового проходження фрагмента програми можна використовувати команди, наведені в табл. 6.1.

Таблиця 6.1

Команди покрокового виконання

Команда	**«Гарячі» клавіші**	**Пояснення**
Step Over (Покроково без заходження в...)	F8	Покрокове виконання рядків програми (виклик функції або процедури вважається за один рядок), входження у функції та процедури не відбувається
Trace Into (Трасування із заходженням у...)	F7	Покрокове виконання програми із заходженням у функції та процедури, що викликаються
Trace to Next Source Line (Трасування до наступного рядка)	Shift + F7	Перехід до наступного виконуваного рядка

Команда	«Гарячі» клавіші	Пояснення
Run to Cursor (Виконати до курсора)	F4	Команда виконує програму до того виконуваного оператора, на якому розташований курсор у вікні редактора коду
Show Execution Point (Показати точку виконання)		Команда поміщує курсор на операторі, що буде виконуватися наступним

Використання на практиці описаних у табл. 6.1 команд проілюструємо прикладом (рис. 6.2).

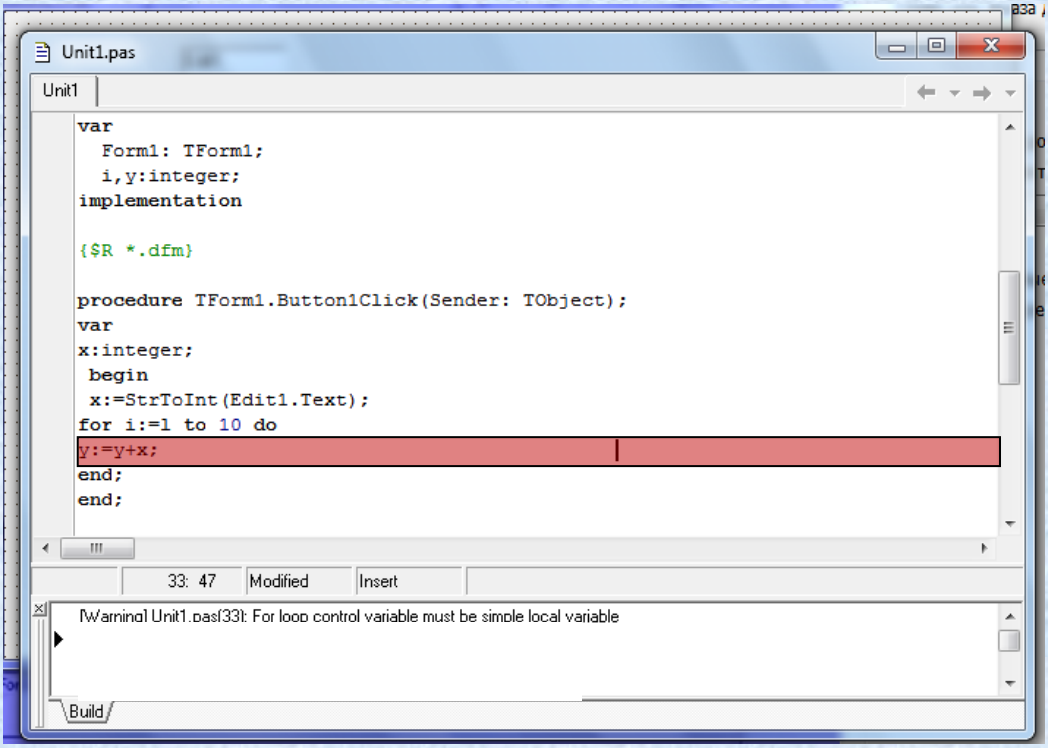

Рис. 6.2. Покрокове виконання програми

Розглянемо потужніший інструмент — введення в додаток точок переривання (breakpoint). Щоб установити точку переривання, досить у вікні редактора коду клацнути мишею на смужці лівіше від коду необхідного рядка. Рядок набуде червоного кольору, і на ньому з'явиться яскрава червона точка (рис. 6.3).

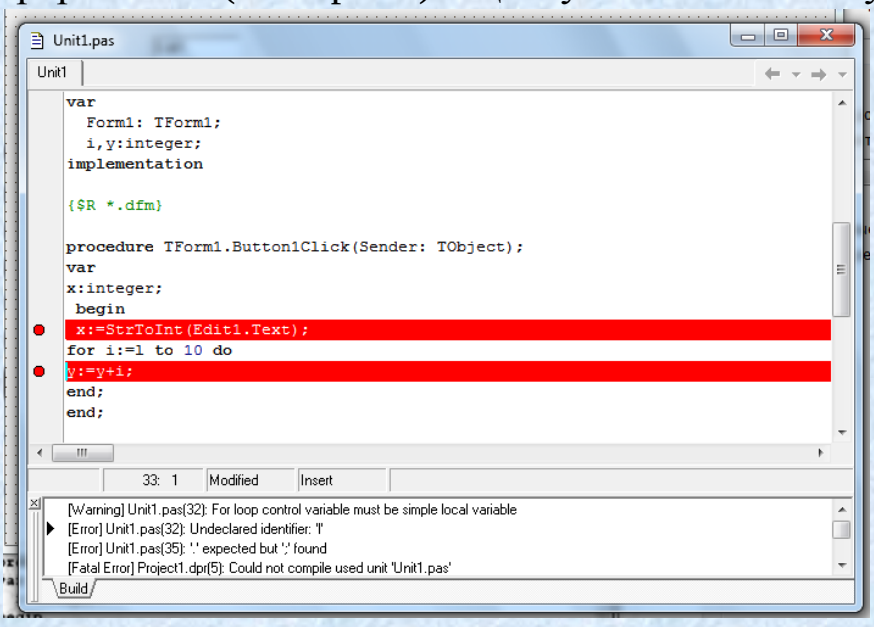

Рис. 6.3. Введення в додаток точок переривання

Якщо тепер ми запустимо додаток на виконання й почнемо з ним працювати, то як тільки керування перейде до рядка, у якому вказано точку переривання, виконання програми перерветься. Таким чином, ми одержали той самий результат, що й у разі описаного раніше виконання програми до точки, вказаної курсором (натискання клавіші **F4**). Однак перевагою введення точок переривання є те, що одночасно можна вказати кілька точок у різних місцях коду й у різних модулях. Додаток буде виконуватися доти, доки керування не перейде до першої зустрінутої в програмі точки переривання. Наступний запуск приведе до зупинки в другій точці переривання і т. д.

Для того щоб прибрати точку переривання, досить клацнути мишею на червоній точці лівіше від коду відповідного рядка. Точки переривання можна встановлювати тільки на виконуваних операторах. Якщо ви, наприклад, спробуєте встановити точку переривання на рядку, що містить оголошення змінної, то в момент запуску додатка в червоній точці, яка виділяє рядок переривання, з'явиться хрестик. Цим Delphi попереджає, що переривання не буде, оскільки оператор невиконуваний (рис. 6.4).

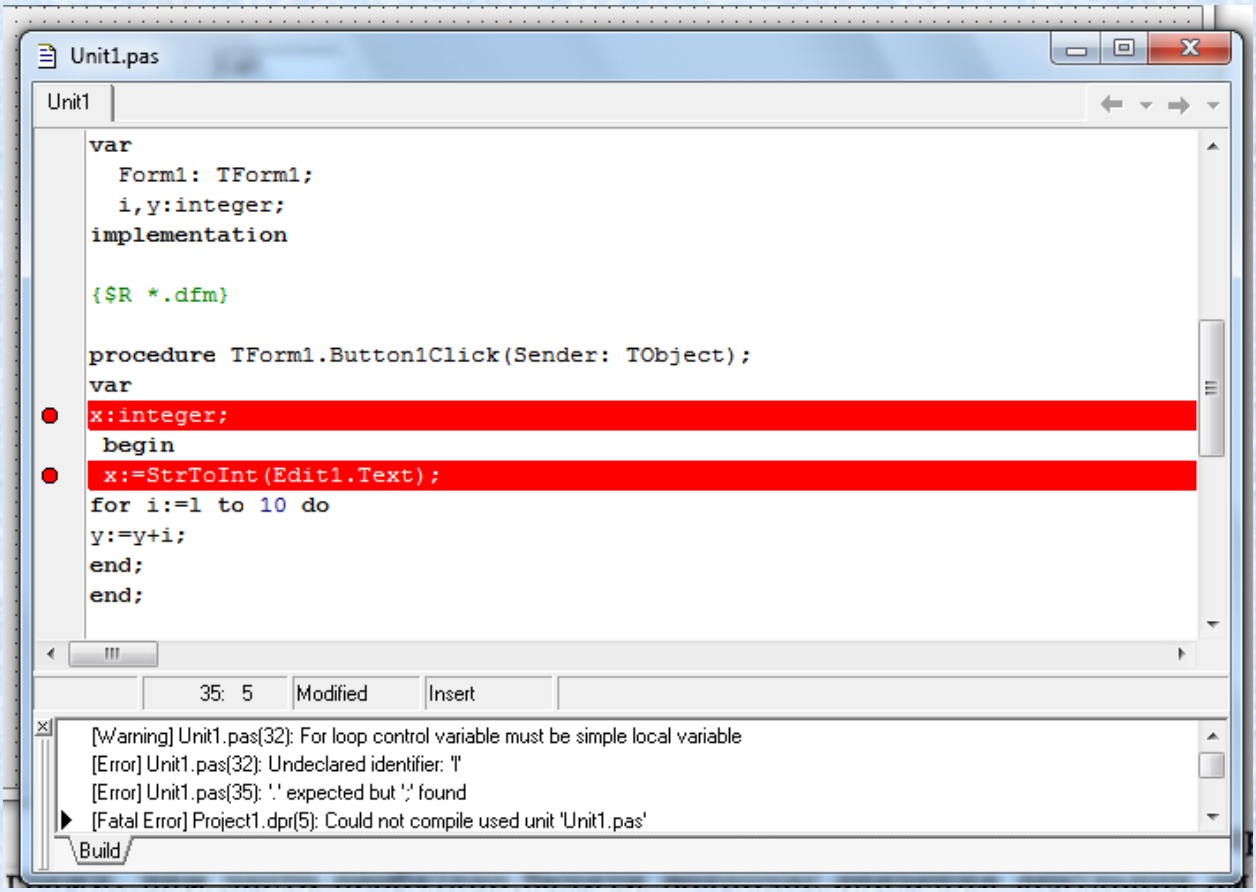

Рис. 6.4. Спроба встановлення точок переривання на виконуваному і невиконуваному операторах

Щоб виправити помилки, які виникають у ході виконання програми, нам часто необхідно бачити водночас значення декількох змінних. Таку можливість дає вікно спостережень **Watches**.

Зробити його видимим можна за допомогою команди **View→Debug Windows → Watches** або підвівши курсор у коді до потрібної нам змінної й натиснувши одночасно клавіші **Ctrl + F5**. При цьому вікно спостережень автоматично відкриється, і в ньому з'являться ім'я змінної та її значення (рис. 6.5). Далі можна підвести курсор до іншої змінної й знову натиснути **Ctrl + F5**, після чого у вікні спостережень з'явиться новий рядок. Понад те, ми можемо виділити курсором якийсь вираз, натиснути **Ctrl + F5** і побачити у вікні спостережень значення цього виразу.

ТЕОРЕТИЧНІ ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ

1. Як можна відкомпілювати додаток у Delphi?
2. Наведіть приклади помилок, які можуть виникнути в програмі.
3. Назвіть команди покрокового виконання програми. Перелічіть їхні відмінності.
4. У чому полягає призначення точок переривання?

Для чого призначене вікно спостережень Watches? Як його викликати?